Packet Tales

by Bill Piazza, KB4QVY

NTS Traffic Handling on Packet

(Reprinted from "White Noise, The Official Newsletter of the Palm Beach Packet Group")

Each day, more and more NTS (National Traffic System) traffic is handled by packet BBS forwarding. You may have seen the commands dealing with traffic on the local BBS and you've probably read a few of the messages as they passed through your area, but you may not have known how to deliver or initiate NTS traffic. Read on...

The first thing to realize about "traffic handling" is that the protocols used by traffic handlers did not spring up over night, but rather are the result of years of "evolution" and they are in widespread use. Because they evolved largely from CW practices, many aspects of traffic handling may seem redundant on packet radio, which leads us quickly to the second point - a very important one: not all traffic handing is done on packet radio. You must remember that if you take steps to "streamline" the handling of traffic on packet, you may make things extremely difficult at the places where traffic is taken off of packet and put into a voice or CW net or vice versa.

A piece of NTS traffic has a standard format which can be broken down into four pieces. First, there is the preamble, which tells how the message is to be handled, how important it is, where and when it originated, etc. Then comes the address, which usually includes a US postal address and a phone number. (Most non-emergency traffic is delivered by phone.) The third section is the actual text of the message and the fourth section is a signature. On packet, in addition to these four basic parts, we need to be concerned about the BBS commands used to handle traffic and also the contents of the TO, @, and SUBJECT fields.

The NTS Message Format

First let's talk about the traditional NTS message. I'll admit right off the bat that I'm not an expert on NTS traffic and will defer questions to the ARRL Operating Manual. But I can explain the basics.

A typical NTS message looks like the one shown in Figure 1 on page 10. The preamble information reads as follows: This is message number 12 (a routine message) which originated at station K4XX in Miami, FL on April 6 at 1830z. The text of the message contains 15 words. The message is to be delivered to John Doe in Southview PA at the address or phone number provided.

The operator who placed this message on the BBS also included his own call and BBS at the end of the message so that he could be reached easily - this is generally a good idea.

Here are some of the variations you might encounter in the preamble or text of the message:

1. The priority of the message may be ROUTINE, WELFARE, PRIORITY, or EMERGENCY. The first three are usually abbreviated as 'R', 'W', or 'P'. The last one is always spelled out in full - 'EMERGENCY'.
2. Immediately after the message number in the preamble, there may be special handling instructions. The special instruction codes start with the letters 'HX':
   - HXA *nnn* - You may call the address COLLECT if you live within nnn miles.
   - HXB *nnn* - Cancel this message if it is not delivered within *nnn* hours of the time it was filed and then send a 'service message' back. (A 'service message', by the way, is simply a message sent back to the originating station of a piece of traffic which tells what the disposition of a piece of traffic was. Service messages should always refer to the original message number and they follow the same format as a regular NTS message. Ordinarily, you will only send this message back to the originator if you could not deliver his traffic or if he requested a service message.)
   - HXC - Please send service message reporting the date and time of delivery to the originating station (another service message).
   - HXD - Send a service message to originator stating when (and from whom) you received the message and when (and to whom) you relayed or delivered it.
   - HXE - The station delivering the message should request a reply from the addressee and originate a message back.

12 R K4XX 15 MIAMI FL 1830Z APR 6

JOHN DOE
123 E MAIN ST
SOUTHVIEW PA 15361
(412) 356-7940

ENJOYED YOUR COMPANY X
HOPE YOU HAD A PLEASANT TRIP BACK X
COME BACK SOON

RALPH

K4XX @ W4NVU

Figure 1. Typical NTS message

- HXF *date* - Hold delivery of this message until the specified date.
- HXG - Delivery by toll call or mail not required. If there is an expense involved in delivering the message, cancel it and service originating station.

3. Commonly used phrases in the text may be replaced by 'ARRL numbered radiograms' so that less information has to be sent over the air. For example, the phrase 'Greetings via Amateur Radio' may appear in the text as ARL FIFTY. (That is not a misprint - we use ARL, not ARRL.) Whenever these 'canned phrases' are contained within the text of the message, the letters 'ARL' will appear before the word count in the preamble.
4. Note that the letter 'X' is used to denote the end of a sentence or thought and that the Xs count as words. Groups of figures, such as numbers are also counted together as words.

Packet BBS Traffic Commands

Many of the commands that you use regularly on packet BBSs have special 'subcommmands' specifically for NTS traffic. For example, you may have used the 'S' command to send a message, or 'SP' to send a private message, but did you know that there is also an 'ST' command for 'sending traffic'? The reason that NTS traffic has its own set of commands is because in some instances NTS traffic is handled differently than standard BBS mail and by using the ST command instead of S you are telling the BBS that this message falls into the category of messages that require NTS handling. Always use the ST command to send NTS traffic.

There is also an LT command (which lets you list only the traffic waiting to be moved by the BBS) and a KT command (which is how you kill a piece of traffic to get it off of the BBS after you've claimed responsibility for it.)

When you send a piece of traffic or list the traffic on a BBS, you will see that just like regular BBS 'mail', NTS Traffic also has a TO, @, and SUBJECT field. There has been much debate over the last few years about what exactly should go into those fields and whether packet routing should be based upon the phone number and area code or zip code of the addressee. A sort of standard has emerged but has not yet been officially 'blessed'. I recommend that you use these fields like this:

- TO -- use the addressee's zip code.
- @ -- use the letters 'NTS' followed by the addressee's 2 letter state abbreviation (e.g., NTSFL, NTSPA, etc.)
- SUBJ -- use 'QTC 1 *city area-code phone-exchange*'. QTC 1 means that this message contains 1 piece of NTS traffic (always send only 1 piece of traffic per packet message, even if you have more than one message going to the same place). By including the city and phone exchange in the subject, you make it easier for the traffic handler on the other end to determine if this message can be delivered in his calling area, without forcing him to read the entire message. You may optionally include the entire phone number.

Using Figure 1 as an example, the command to send it and the subject entered would look like this:

```
>ST 15361 @ NTSPA
Enter subject for message 1634:
QTC 1 SOUTHVIEW 412 356
```

After this, you enter the message in the traditional format.

Delivering Traffic

OK, so you understand a little bit about how NTS traffic gets onto the BBS... what about getting it off? Many pieces of traffic come into the Boca, Boynton, West Palm Beach, and Pompano areas to be delivered, and you can deliver them! Here's all you need to know:

1. When you log onto the local BBS, you can use the LT or L commands to find the traffic waiting on the BBS.
2. If you find a piece of NTS traffic waiting to be delivered in your area, go for it! Don't be afraid... it is really simple:
   a. If you decide to deliver a piece of traffic, get set up to save a copy to diskette or printer (how you do this will vary depending upon the type of computer and terminal program you are using) and then read the message (by using the BBS 'R' command). In my opinion, saving a copy on diskette or the printer is VERY important - you want to make sure that you don't loose the message before it is delivered and copying it down by hand can lead to errors.
   b. After you download it and have accepted responsibility for it, use the KT command to kill the message. Do this before you disconnect from the BBS and deliver the message. If you come back later to kill the message, you may find that someone else has read the message off of the BBS and also decided to deliver it (and the message was therefore delivered twice) or that the BBS forwarded the message to a closer BBS and it is not there for you to kill!! (If you saved a copy of the message on diskette, close the capture file before you Kill the message. FP&L may erase the message for you if you don't.)
   c. Once you accept responsibility for a message, BE PROMPT! Hams look a little foolish delivering a message that is four weeks old and it does nothing to enhance the reputation of packet radio as a reliable means of communicating. Deliver the message at a reasonable hour, but do it the first chance you get.
   d. When you accept a message and kill the original, you are making a solemn oath to do your best to deliver the message. Under no circumstances let a piece of NTS traffic die from neglect. If you find that you cannot deliver a message, either put it back into the system (via packet, CW, or a Voice Net) or cancel the message by servicing the originator and explain why you could not deliver the traffic - the addressee has moved, disconnected their phone, etc.
3. When you deliver the message, explain briefly who you are and how the message arrived in your hands. Then read the text of the message and the signature and ask the recipient if they'd like to reply using the same amateur radio traffic network. Make sure that they know that there is no charge involved.
4. When accepting messages to place into the NTS system, try to convince the sender to keep them brief - 25 words or less is ideal. Long messages move through the system slower and have a higher probability of picking up erroneous changes if moved on CW or Voice, where a human operator will be copying the message down and reading it to someone else. If the sender includes text similar to a numbered radiogram, you may want to suggest that they use the 'canned message' in place of it to keep the word count low.

In closing, let me encourage you once again to try some NTS work. It can be a lot of fun and it is certainly valuable during emergencies. You can practice on a relative or friend by composing a brief message for their birthday, anniversary, or just to say 'Hi' and placing it on the local BBS as described above. Let's help get the BBS Sysops out from under the NTS crunch by pitching in and delivering traffic in our hometowns.

73 Bill KB4QVY @ WB4TEM

Packet Tales

by Bill Piazza, KB4QVY

An interview with Mike Chepponis, K3MC

(This month, I am presenting an interview with Mike Chepponis, K3MC. Mike is the original author of special firmware for TAPR 2's and clones called "KISS" - Keep It Simple Stupid. Mike, along with Phil Karn, KA9Q, and a number of other packeteers around the country, are proponents of a networking protocol called TCP/IP. This protocol is modelled after the Defense Department's ARPANET and the reason for the KISS EPROM was to allow the TCP/IP software running on a personal computer to have full access to and control over the AX.25 link layer. While their efforts were originally oriented towards the TAPR 2 TNC and IBM PC clone computers, TCP/IP and KISS now run on a number of other computers and TNCs. Some TNC manufacturers (e.g. Kantronics) have even implemented the KISS interface in their standard product line.

Mike first received his Novice license in 1966 at age 11 and upgraded to Extra class a few years later. While attending high school at Culver Military Academy, Mike started a radio club and was exposed to his first computer - an IBM 1401. He wrote a large number of programs during that 4 year period, including a compiler for a "C-like" language called Simpletran (C wasn't invented yet).

At college (MIT), Mike participated heavily in W1MX, the ham radio club station, was station manager, and received his degree in Electrical Engineering and Computer Science. In the late 70s, he got interested in repeaters and control systems for repeaters. At Dayton, in 1984, Tom Clark, W3IWI, convinced Mike to put up a BBS which was the first BBS serving Pittsburgh, Pennsylvania.

Mike now lives in the bay area of California.

I met Mike in Pittsburgh in late December, 1987, while we were both back visiting the old stomping grounds. I asked Mike if he'd be interested in doing an interview and he agreed. The interview was actually conducted using packet BBS mail forwarding between Florida and California during the months of January through March, 1988)

KB4QVY: As you know from our discussion in Pittsburgh, I'm a NET/ROM backer. That's mainly because we found that it is allowing us to fulfill many of our goals down here (in Florida). I'd like to know a lot more about TCP/IP, as I'm sure that others down here would, but it's hard to learn about Paris from some place in Kansas, if you know what I mean. Can you start by telling us a little about TCP/IP?

K3MC: First, let me clear away some confusion: NET/ROM, Texnet, and COSI **(Ed. note - now called ROSE)** are all fundamentally different from TCP/IP. TCP/IP can run on "top of" any of these lower layers. TCP/IP encompasses much more than just a networking layer (like NET/ROM or COSI), or a partial transport layer (like NET/ROM). To be sure, the COSI gang promise lots of programs, all based on the IOSRM (pronounced Eye Sore-emmm) CCITT protocols. These protocols are all virtual-circuit oriented, and very few implementations exist of anything other than the Link layer. It has been shown that virtual circuits are a Bad Way to build a network that is plagued by unreliable links; this is why, for example, NET/ROM uses a datagram-based protocol as its network layer, so it can recover from node failures. In the COSI model, if a link goes down, so does your connection! In TCP/IP, this is not the case: when a link goes down, you can change the routing and the packets keep flowing! Or, one can simply wait until the link comes back up. (TCP/IP is very forgiving about these sorts of things; it uses something called "Polynomial Backoff" to help keep the channel from being pummeled into the ground.) Once I had a file transfer going to a station through a digi, but the digi kept going down. Well, after two days, the digi was up again, and the file transfer continued to completion!

TCP/IP means "Transmission Control Protocol/Internet Protocol" and is the name given to the DARPA Protocol Suite (set of protocols). This protocol suite consists of a number of protocols, but those that are used in ham radio include UPD (User Datagram Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol), and ICMP (Internet Control Message Protocol ffla part of IP, really"), Telnet (A host-to-host communication method, used in ham

radio as a keyboard-to-keyboard service) as well as TCP and IP. IP is the Networking protocol, TCP is the Transport protocol, and SMTP, FTP, and Telnet combine the Application, Presentation, and Session layers. Thus, TCP/IP implements "all" of the layers of the 7-layer ISO Reference Model.

KB4QVY: Why do you think TCP/IP is a real good deal for the VHF packet network?

K3MC: Well, TCP/IP as used in ham radio is "exactly" the same as that used on the DARPA Internet (the DARPA Internet is sometimes called the Arpanet; DARPA means Defense Advanced Research Projects Agency, a branch of the DoD). The DoD uses TCP/IP because it is a very robust protocol suite, and is likely to still work in times of national emergency (or, in the ham's case, disasters that require ham radio's renowned communications abilities). Plus, we hams, with TCP/IP, get the benefit of years of work on protocols done by the DARPA researchers. TCP/IP specifications are all publicly available, and in fact, can be acquired from TAPR at nominal cost on IBM PC-compatible floppies. In addition, TCP/IP is available for hundreds of computers, from dozens of manufacturers and software houses. It is "the" networking standard!

KB4QVY: Earlier you mentioned "Polynominal Backoff" - can you explain what that is?

K3MC: Polymnomial Backoff first retries, say, in 1 second, then 2 seconds, then 4 seconds, then 8 seconds, etc. Thus, when there is intense channel activity, TCP/IP "backs off" to offer less traffic to the channel. When the channel loading reduces, TCP/IP senses this, and continues to do the rest of the transfer "full speed".)

KB4QVY: What has been your exposure to ARPANET?

K3MC: I've been a user of the Arpanet since 1981, when I worked in robotics at Carnegie-Mellon University. I used the old NCP, and later started to use the new-fangled TCP/IP, that replaced NCP, and fixed quite a few problems with the old NCP. I've been an active user of the Arpanet ever since.

Another big user of the Arpanet is Phil Karn, KA9Q. It is his vision and persistence that brought the advantages of big-networking philosophy to ham radio. Mark my words, history will look back on Phil's contribution as a turning point in ham radio packet, maybe even ham radio in general!

KB4QVY: From the reading that I've done, it seems that ARPANET has had it's share of problems in the past. For example, I remember reading that ARPANET has, on more than one occasion, "deadlocked" when two adjacent switches had full buffers which they could only empty by passing frames to each other. Another problem that I remember reading about is that when one area of the network becomes congested, it can impact the performance in a remote area of the network which has nothing to do with the congested area, other than the fact that packets destined for both of these areas go through a common switch. These problems were both related to the intentional lack of a link layer flow control mechanism in ARPANET, if I remember right. In Pittsburgh, you stated that lack of flow control was an asset. Can you explain?

K3MC: Yes, I know about several times the arpanet became congested, and even catatonic. The catatonic episodes were due to bugs in the gateways, and the poor performance was a result of improper load balancing. When BBN tuned the gateways, things returned to the normal, superior service we are all used to. The other thing BBN did was to analyze where the traffic was going, and to recommend that new gateways, using alternate routes, be installed to do a better job of load sharing. I suspect that we will need to do the same sort of thing in our budding AMPRNET. When we find, say, a huge amount of traffic flowing between Northern and Southern California, we will need to install more IP switches (gateways) to handle the traffic. This is nothing magic about this.

As to an intentional lack of a link level flow control mechanism, I "do" think that the gateways (using 1822 protocol) have some sort of flow control. But even if they didn't, proper implementations of TCP/IP, like Phil's package, automatically "back off" when acks are slow in coming. Phil measures the "round trip time" - that is, the time from sending a packet to getting its ack. If a packet is not acknowledged within somewhat longer than the round trip time, the packet is resent. Now, if "that" packet is not acknowledged, it waits two round trip times, etc., backing off using a provably-superior Polynomial Backoff algorithm, which offers less and less traffic to the network. This allows the

network to "flush" itself, and get back on its feet. The individual TCP/IP nodes automatically adjust their retransmission timers so as not to overload the network.

Throughput may suffer, but the net won't "crash". I think that the experience of the arpanet has taught us many things, and I don't think we will be committing the same mistakes. You see how wonderful it is to use protocols that are already in widespread use: one can take advantage of that huge pool of experience and accumulated wisdom!

About my comment about lack of flow control, it is likely I said this in connection with a KISS TNC. There, packets come in and, after CRC checking, are sent immediately to the host. The host must receive the packet at that time. You see, there have to be buffers "somewhere" in the system, and keeping them closest to the application (TCP/IP in this case) makes better sense than keeping the dangling bits on the TNC with flow control. (Incidentally, the KISS TNC couldn't care less about the rate you shove bits AT it (up to about 38.4k baud); it isn't doing much else, anyway!) Flow control is required by applications that are improperly implemented.

KB4QVY: Can you highlight some of the features of TCP/IP that NET/ROM cannot match, such as simultaneous file transfer and keyboard operation, etc. (Go ahead, tell me what's possible with a TCP/IP based digital network and why nothing else would do... but let me warn you now that if the reason nothing else would do ties in too much with simple transmission speed I won't buy it... hi hi).

K3MC: Comparing NET/ROM and TCP/IP is like comparing a WW I biplane to an F-16. Why this is so will require another message...

For a short reply, suffice it to say that more and more people realize that TCP/IP is the New Wave of networking. And there are more surprises coming...

Since we last communicated, Ed Frank, W9NK has integrated net/rom into TCP/IP! So now, TCP/IP nodes can - at the same time - be net/rom nodes, just like any other net/rom node. PLUS, it can pass datagrams right on top of net/rom layers, AS AN END NODE! This means that no "connect to the local net/rom node" is required! Because TCP/IP is now an end net/rom node (it is - effectively - permanently "connected"!)

KB4QVY: In Pittsburgh, you made some interesting comments to me about the packet network and keyboard users. I think you either said that the network was not suited for keyboard users or you could not understand why people would want to use packet for keyboard QSOs...yet hams have been doing that for years on RTTY, and most recently AMTOR and packet. Can you clarify what you said and what it really meant?

K3MC: What I said was that I could not understand why folks would use packet for kbd-to-kbd QSOs. Now, perhaps this is chauvinism coming from the arpanet side of me, where kbd-to-kbd "contacts" are extremely rare, rather than from ham radio side, where, as you've pointed out, RTTY and AMTOR, and now packet are being used kbd-to-kbd. I started out on "other" digital modes (other than CW, that is) using RTTY. In fact, my first TU was a 5763 one-tube job from an old handbook that had to be tuned to the space frequency (only!) to function. It was fun "talking" to other hams, and I guess I considered it an extension of CW. It was also really good for copying the ARRL bulletins.

In this day and age, though, RTTY is essentially obsolete. Common ham RTTY, using 170 Hz shift FSK, is neither spectrally efficient nor does it have particularly good error characteristics, especially with regard to Eb/N0. AMTOR, an ARQ protocol, does much better in the HF environment. For an excellent discussion of better ways to do HF data transmissions, see the two-part article by Barry McLarnon, VE3JF, in December 1987 and January 1988 QEX.

But more to your question: Why even "do" kbd-to-kbd? Well, I'm not sure... For instance, if we were in real-time communications with these questions, you'd say something, and then I would respond, and then I'd ask questions, etc., and we'd have a kbd-to-kbd QSO. But, how much better it is to get a message from you, and let me respond when I can think about an answer, to go back and review what I'm going to send you, to fix the grammar and spelling errors, to see if it all logically hangs together, etc! Plus, you could be sleeping or out shopping or working DX on 12 meters or something while I am, at the same time, composing a reply. Basically, messages eliminate the tyranny of real-time, johnny-on-the-spot.

shoot-from-the-hip kind of replies. It is a way of freeing ourselves from having to "be there" to get information transferred. This was one of the original reasons for BBSs, to be sure...

From what I have seen, most kbd-to-kbd contacts over VHF packet use one or more digipeaters or NET/ROM nodes. That is, direct kbd-to-kbd contacts are rare, because if it were possible just to dial in the local repeater and "talk" to the other station, that is surely a faster method of getting the information across.

So, most kbd-to-kbd contacts happen over one or more intermediate links. Presumably, no repeater is available to use over the great distances covered by the intermediate links, so, in effect, kbd-to-kbd is the "only" means of carrying out a real-time QSO. It seems that in this case, kbd-to-kbd QSOs are reasonable. But are they, really? Wouldn't it be just as easy to send a message to the BBS near the other station? Then you wouldn't have to deal with retries, link timeouts, channel activity, whatever... it would be done automatically for you, by computers. This is the essence of networking.

Now, I wasn't saying that any advanced network would not be capable of handling kbd-to-kbd QSOs, I was just wondering why anybody would want to do that. The network that many of us envision would enable direct digital voice communications between any two points in the US, and eventually in the world. With voice as an option for real-time (interactive) QSOs, I don't see any need for kbd-to-kbd QSOs. Perhaps you should ask somebody who does this why he does it.

KB4QVY: When we were using only the AX.25 link layer and a string of digipeaters, it took forever to recover from collisions. The really nice thing that NET/ROM offered was hop-to-hop acknowledgement (or, as Software 2000 calls it, "store and forward" packet switching). Yet, from what I understand about TCP/IP, IP switches provide only end-to-end acknowledgement, which sounds suspiciously like what we are trying to get away from. How would the performance of TCP/IP compare to straight AX.25 on a really long haul with lots of collisions?

K3MC: Both NET/ROM and TCP/IP offer hop-by-hop acknowledgements. In TCP/IP parlance, you can choose either "high reliability" or "low delay" in your IP options. "High reliability" means to use hop-by-hop acks, and "low delay" means do not use hop-by-hop acks. TCP/IP lets you choose, depending upon the circumstances.

When "high reliability" is used, underlying AX.25 connections are set up between the TCP/IP nodes, and the link layer does acking (in addition to the higher-layer acking done by TCP). TCP/IP automatically fragments IP datagrams into segments of appropriate size for transmission by AX.25 LAPB mode. Normally, however, TCP/IP nodes run in "low delay" mode, where IP datagrams are encapsulated inside AX.25 UI frames (i.e., "beacons"). This results in quite good throughput, with a minimum of overhead information.

But the way you phrased your question leads me to believe that I must explain how we intend to build large TCP/IP networks. Firstly, as you have observed, having everybody on one frequency is a poor network strategy. Here in Northern California, we are planning on breaking up the various pockets of activity into "cells" of perhaps a dozen or so stations. Each station within a cell will be able to easily work the local IP switch. There will be IP switches for perhaps a dozen cells in Northern California. These IP switches will be dual-ported, with one port being a 1200 baud port, and the other being a 56k baud port. Low-speed users can come in on the 1200 baud port and be switched over the network at 56k baud to their intended destination. Those of us with 56k baud modems at home will be able to enter the network at that speed, and use the network resources at that speed.

For the low-speed users, their connections to distant TCP/IP nodes would appear virtually instantaneously.

The important points are: 1) The 56k baud network will be near collision-free; it will never have more than 2 56k baud nodes on a given channel and 2) Limited CSMA/CD channels, like in the local cells, will ensure that the problem with collisions and hidden transmitter problems will be bounded and unlikely. We would like to eventually move to the scheme outlined by Phil Karn in the 29 Aug 87 ARRL Networking Conference, where each switch has a single transmitter on a single frequency, and multiple receivers, each on their own frequencies. The receivers are tuned to the adjacent IP switches and beam antennas are used; the IP switch transmitter would use an omni

antenna. This scheme is *completely* collision-free.

KB4QVY: Mike, I mentioned to you that I believe in competition and "constructive brick throwing" as a way to understand both sides of an argument, but I also believe that we have to keep the discussion factual and above a personal level. Recently, the discussions of TCP/IP vs NET/ROM have sometimes been reduced to mere name calling with no factual content.

For example, in a recent issue of Grapevine (newletter of the Georgia Radio Amateur Packet Enthusiasts), one of the Georgia guys (I think he's the editor of the newsletter) called us a bunch of cheap, appliance operators because we're using NET/ROM. I think attitudes like that serve only to alienate and polarize, and they do nothing for the advancement of packet radio. That's one of the reasons why I'd like to do these interviews - to try and get objective, factual information to all interested parties on both sides...

K3MC: Let me address the zealousness of some TCP/IP converts. Those of us that have been using the Arpanet for years are pleased that ham radio is finally getting into the '70s, as far as networking goes, with TCP/IP. Some of us lose sight of the thousands of hours others have invested in hardware/software to do a perceived function, namely mail forwarding, and how we (TCP/IPers) view that system as not useful for more advanced networking concepts. But what a long, strange trip it's been! Now we have the basis for truly remarkable networking at our fingertips, and sometimes some of us can't comprehend why anybody could endure using anything else. Call it a lack of patience ;-) But be certain: we firmly believe that TCP/IP will become the dominant world-wide protocol suite of choice in the next couple of years. Sometimes we just don't understand why other folks don't see it this way, too! Try to be forgiving if some of us sometimes downplay current ham packet radio accomplishments.

Anything else you'd like to explore?

KB4QVY: Mike, I think we've done enough damage for one newsletter, hi hi. I'd just like to thank you for taking the time to share your point of view and all.

73 Bill KB4QVY @ WB4TEM

K3MC: Best -Mike